A TRIDENT SCHOLAR
PROJECT REPORT

NO. 233

"Reactor Noise Analysis"

19950927 145

UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MARYLAND

REPORT DOCUMENTATION PAGE		Form Approved OMB no. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 9 May 1995	3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE Reactor noise analysis

5. FUNDING NUMBERS

6. AUTHOR(S)

John C. Ventura

7. PERFORMING ORGANIZATIONS NAME(S) AND ADDRESS(ES)

U.S. Naval Academy, Annapolis, MD

8. PERFORMING ORGANIZATION REPORT NUMBER USNA

Trident report; no. 233 (1995)

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

Accepted by the U.S. Trident Scholar Committee

12a. DISTRIBUTION/AVAILABILITY STATEMENT

This document has been approved for public release; its distribution is UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words) Noise analysis experiments were conducted on the TRIGA-F reactor at the Armed Forces Radiobiology Research Institute at Bethesda, Maryland. A pseudo-random reactivity perturbation was inserted into a cold critical core. Power level fluctuations were monitored and an estimated transfer function was constructed. From graphical plot analysis, the breakpoint frequency corresponding to β/l was measured to be 183 (units) 1/sec. This is the first such measurement on a TRIGA-F reactor by forced oscillation and is accurate to accepted values.

14. SUBJECT TERMS nuclear reactors; noise analysis; transfer function

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED

NSN 7540-01-280-5500

Standard Form 298 (Rev.2-89)

"Reactor Noise Analysis"

by

Midshipman John C. Ventura, Class of 1995
United States Naval Academy
Annapolis, Maryland

John C Ventura

Certification of Adviser Approval

Capt Matthew J. Forsbacka, USAF
Department of Physics

Matthew J Forsbacka

09 May 95

Acceptance for the Trident Scholar Committee

Associate Professor Joyce E. Shade
Chair, Trident Scholar Committee

Joyce E Shade

9 May 95

Reactor Noise Analysis

Noise analysis experiments were conducted on the TRIGA-F reactor at the Armed Forces Radiobiology Research Institute at Bethesda, Maryland. A pseudo-random reactivity perturbation was inserted into a cold critical core. Power level fluctuations were monitored and an estimated transfer function was constructed. From graphical plot analysis, the breakpoint frequency corresponding to β / l was measured to be 183 (units) 1/sec. This is the first such measurement on a TRIGA-F reactor by forced oscillation and is accurate to accepted values.

Keywords

Nuclear Reactors
Noise Analysis
Transfer Function

Accesion For		
NTIS CRA&I	☑	
DTIC TAB	☐	
Unannounced	☐	
Justification		
By		
Distribution /		
Availability Codes		
Dist	Avail and / or Special	
A-1		

The author would like to thank the following

for their invaluable support and assistance:

Reactor Noise Analysis

Reactor Noise Analysis

I. Objective

This project sought to use noise analysis to quantify dynamic parameters within a nuclear reactor operating at steady state. In particular, the dynamic parameter β / l was measured. This parameter, which will be defined in detail later, is useful because it can be used as a deterministic, or causally predictable, measure of the isotopic content (Uranium-235 vs Uranium-238) of the reactor fuel.

One military application of this research could relate to the North Korean reactor situation. It would be possible to verify the usage logs of North Korean reactors with the actual isotopic content of the fuel. In the event that they were manufacturing weapons grade Plutonium, their core's β / l would show evidence of fuel swapping, a necessary step in the plutonium manufacture process.

Noise analysis techniques involve constructing a transfer function for the reactor as a "black box" system. A transfer function is a means of considering a complex system, such as a nuclear reactor, to be a single operation that modifies an input signal to produce an output signal. Developing a transfer function is possible from an observation of the output power level of the reactor in response to a known input reactivity oscillation.

II. Background

A nuclear reactor is a stochastic system. Individual events do not necessarily lead to predictable individual effects on the atomic level. In other words, the system is random on a microscopic scale. This means that even when a reactor is operating at "steady state" power, it is actually making small fluctuations about an average power

level. To better understand this principle, a discussion of the processes involved is useful.

The specific reactor used for this experiment was the Training Research Isotope reactor manufactured by General Atomic (TRIGA) located at the Armed Forces Radiobiology Research Institute (AFRRI). This reactor is a thermal reactor, meaning that it primarily uses neutrons in thermal equilibrium with their surroundings to carry on a chain reaction. It should be noted that on an atomic scale, these thermal neutrons are often considered to be slow moving. This is in contrast with high energy neutrons, called fast neutrons, which are the prime movers in a "fast" reactor.

An important term in discussing the chain reaction is "critical." A critical reactor is one which is maintaining a stable power level, ie. the neutron population is remaining at a constant size.

The neutron chain reaction process is modeled by a "neutron racetrack". This is depicted in Figure 1. At any given point in the cycle, neutrons are present in the system. Thermal neutrons have a high probability of colliding with atoms of Uranium-235. Possible reactions after collision include absorption, scatter, and fission. If the uranium fissions, fast neutrons, fission products, and gamma radiation are emitted. There is a randomness introduced in this process, as the number of neutrons released per fission is not constant but varies between 1 to 5. These neutrons, which are immediately released, are known as prompt neutrons.

There are several events which can then happen to these new neutrons. The first and most common of these is a series of elastic collisions with hydrogen in the fuel and in the water coolant. These collisions slow the neutron. This slowing down process introduces a random factor dependent upon the number and type of collisions made.

Assuming a neutron slows to thermal velocities without being lost, it can then strike another U-235 atom, causing the fission process to repeat. The second possible set of events are the ways in which neutrons may be removed from the fission chain. These are:

- Neutron leakage from the reactor core. These neutrons are lost to the surrounding environment.

- Fast neutrons can also be absorbed by U-238 that is present in the core, effectively removing them from the process.

- Neutron are absorbed by poisons in the core. (eg. The control rods are designed to absorb neutrons.)

There is another source of neutrons to carry on the chain reaction, namely the decay of certain fission products. These fission products, called delayed neutron precursors, decay by releasing a neutron. They are categorized into 6 groups by how long it takes for them to give off a delayed neutron. Because these time delayed neutrons are required to maintain the self-sustaining chain reaction, fission reactors are controllable.

The fraction of all neutrons present that were born from delayed neutron precursors is given the name Beta. Because delayed neutrons are born at lower energies than prompt neutrons, they are more likely to reach thermal energies without being lost. This makes them more effective at causing fissions. β is therefore modified to account for this, the result is called "β(effective)". For scientific purposes, it is "β(effective)" that is actually measurable, and the qualifier "effective" is not used in this paper.

To vary the reactor's fission rate, and thus change the reactor's power level, the rate of absorption by non-fuel material is varied. This is accomplished with control rods made of boron or other neutron absorbing material that may be introduced into the reactor core. By controlling how much of the rod is in the core, one controls how many of the neutrons are being absorbed. Less rod in the core leads to less neutrons being absorbed and therefore a higher power level. With the reactor critical, ie. at a stable power level, withdrawing rod will cause the reactor to increase power, then come critical again. The amount of rod withdrawn is measured in reactivity, which has units of dollars and cents. Reactivity is a measure of how close the reactor is to being critical. For example, assume a reactor is sub-critical by 55 cents. When a rod in this reactor is withdrawn 5 cents, the reactor becomes 5 cents closer to critical, and is now only subcritical by 50 cents.

The average time between the birth of a neutron and the removal of that neutron

from the system, either by absorption or leakage, is l, the mean neutron lifetime. Both β and l vary depending upon the composition of the fuel core. As the core ages, there is a loss of U-235 and a buildup of poisons. This leads to a shortened neutron lifetime because neutrons are absorbed more quickly after their birth. β increases with the age of the core. Because there are more poisons in the core, prompt neutrons encounter more poisons while slowing. This makes delayed neutrons more effective as they do not experience these poisons for as long because they are born at energy levels closer to thermal. Since β increases and l decreases, β/l must increase with the age of the core.

III. Theory

As has been discussed, there are a number of seemingly random factors involved in the chain reaction process. This randomness makes it impossible to predict what any given neutron will do at any given time. Likewise, it is impossible to track a neutron's heritage back through the reactions. It is, however, possible to examine the reactor on a macroscopic scale with the technique of noise analysis.

The most significant of the random factors is the fact that the number of neutrons liberated per fission is not constant. In a fast reactor, this is the only major source of noise. This makes noise analysis on fast reactors a straight forward process (Uhrig, 1970). It is important here to clarify that the reactor noise which is being discussed is not acoustic noise, but rather the tendency of the power level to vary with seemingly random oscillations about a mean value while at a "steady state" condition. The oscillations themselves are the noise.

In a thermal reactor, the random factor of thermalization, or slowing down, of fast neutrons adds another dimension of noise to the power output. This additional noise blurs the original neutron production noise (Uhrig, 1970).

This extra random factor made noise analysis on thermal reactors extremely difficult. Noise analysis has been done for decades on fast reactors (Thie, 1981). Only recently has data acquisition and analysis technology progressed to the point where it is possible to conduct these experiments on thermal reactors without major difficulties. This is now accomplished by introducing a known input perturbation, in this case

reactivity, which has the effect of re-correlating output noise.

A set of equations exist to define the state of the reactor in terms of the neutron density, which is directly proportional to the power level and the precursor density. These are known as the point kinetics equations. They are given below:

$$\frac{\partial n}{\partial t} = \frac{\rho - \beta}{l} \cdot n \; + \Sigma \lambda_i C_i \; + \; q$$

$$\frac{\partial C_i}{\partial t} = \frac{\beta_i}{l} \cdot n \; - \; \lambda_i C_i$$

Where:
n = neutron density
ρ = reactivity insertion
β = delayed neutron fraction
λ_i = decay constant of ith group of neutron precursors
C_i = Concentration of ith group of neutron precursors
q = independent source of neutrons
t = time
l = mean neutron lifetime

Note that this is a set of 7 equations. The first equation is for neutron density. The second equation relates to the precursor population and is really 6 equations, one for each group of delayed neutron precursors.

β, as discussed in the theory, is one of the important process variables in the production of delayed neutrons.. Direct measurement of this parameter is difficult, even with noise analysis. This experiment was designed to measure β indirectly. The neutron lifetime, l, is another important parameter. The ratio, β/l, is readily measurable from noise analysis, as will be shown.

Noise analysis techniques involve observing the varying output of the reactor. Left undisturbed at a critical power level, the reactor will make small random oscillations about that power level. It is possible, but difficult to analyze this inherent noise, as in the Bars experiment (Bars,1967, Thie, 1981).

A much better technique, both in terms of accuracy and simplicity of experimentation, is to measure excited noise. Because the noise is question is power level noise, and power level is dependant on reactivity, an input reactivity change will affect the power level over time. If a small known reactivity perturbation is input into the system, the power level output will contain time varying noise about the original power level. The oscillations must be kept small to allow for a linear approximation to the point kinetics equations (Moore, 1958).

This power level can be considered to be the output of the system in response to the reactivity input. The Laplace transform of the output divided by the Laplace transform of the input is called the transfer function of the system. (A Laplace transform takes data that is a function of time and transforms it into a function of frequency in what is known as "The Laplace Domain".) Because this transfer function will be in response to a given input, which will be a known perturbation, uncorrelated noise effects will vanish.

A state of the reactor known as cold critical is specified as steady state stability at a low power level. For the purposes of noise analysis, the power must be kept stable and low enough to avoid heating the fuel. The stability requirement ensures a stationary signal, one with a constant average value. This is necessary for the mathematical assumptions involved in the Laplace transform. Heating must be avoided because it would lead to other random noise perturbations being added from thermal motion effects and coolant flow effects.

The transfer function for a reactor system at cold critical is well defined. Appendix 1 is a derivation of the transfer function from the point kinetics equations. The final result is a transfer function, G(s):

$$G(s) = \frac{n_o}{\beta \cdot \lambda} \cdot \frac{(s/\lambda + 1)}{s\left(\dfrac{s}{\beta/l} + 1\right)}$$

Where: All terms are defined as above and

n_o = initial population of neutrons

s = Laplace frequency variable

This transfer function can be analyzed by linear approximation. This technique indicates that the plot of Magnitude vs Frequency will have two corner frequencies. These will be at λ and at β / l. The latter of which is of immediate concern. Figure 2 is a theoretical plot of the transfer function, indicating the two corner frequencies. These corner frequencies appear as the two major inflection points on the plot.

IV. Methods

Phase I. Exposure Facility and Detector Selection

The exposure facility chosen for this experiment was AFRRI's Exposure Room #2 (ER 2.) This exposure room was chosen because it is possible to slide the movable reactor core to within 1 inch of the room itself. Figure 3 is a overhead view of the sliding core assembly. This mobility means that the core can moved far enough away from the room to work, but then brought close enough to measure neutron density from neutrons leaking out of the core, into the room. Another feature of ER 2 is that it has only one half of an inch of aluminum shielding between the core tank and the experiment. This allows all speeds of neutrons to pass through.

The other choice of exposure room would have been ER 1, which has a cadmium shield over the tank projection. This shield absorbs thermal neutrons and allows only fast neutrons to pass into ER 1. Cadmium absorbs thermal neutrons, as does boron and borated polyethylene. These are the shielding materials of choice for slow neutrons.

This experiment was concerned with delayed neutron precursors and delayed

neutrons. Most delayed neutrons are born only slightly faster than thermal, as compared to prompt neutrons which tend to be born fast. Therefore, it was necessary to measure the thermal neutrons, and necessary to use ER 2.

The next step of this experiment was to devise a means to accurately and precisely measure the power level of the reactor. The TRIGA reactor control console has three independent power readouts from three different neutron detectors in the core itself. These power readings were not precise enough for the purposes of noise analysis. To perform noise analysis, it is necessary to have the reactor extremely stable at cold critical power levels. This means a power level of less than 100 watts, or less than 0.01% of the reactor's maximum steady state power level. At this low range, the error in the power level readings is greater than the noise signal that is the desired output.

A Compensated Ion Chamber, (CIC), was chosen to meet the needs of this experiment. This is a device which can discriminate between gamma radiation, and neutron radiation, which is the radiation of interest. With the detector exposed to the core face on in ER 2, and shielded by borated polyethylene on its other sides, it was then necessary to set operating voltages for the CIC. An ion chamber works by detecting ionizing radiation. Neutrons are not ionizing radiation, but gamma rays are. To correct for this, the inside of an ion chamber can be lined with boron. A neutron collides with the boron, releasing an ionizing alpha particle. (Knoll, 1979)

An ion chamber can measure ionizations produced by both alpha particles and gamma rays produced by the core. At cold critical, the gamma to neutron ratio is extremely high. Again, this can be corrected for experimentally. The compensated ion chamber has two equal sized inner chambers. One chamber is boron lined, making it sensitive to neutrons and gamma rays, the other is not boron lined, making it sensitive to gamma rays only. The two signals are subtracted, leaving a signal that is proportional to neutron density only. This is accomplished by reversing the high voltage to the chambers, in fact, by making the two voltage sources of different magnitudes, one chamber can be made smaller than the other for logistical purposes. Voltage lines into the exposure room were fed through a snaked conduit. This prevented harmful radiation from leaking out into the preparation area from the exposure room.

Ion chambers produce a current signal in proportional response to ion pair production. This is accomplished by drawing one half of each ion pair to a high voltage line, then measuring the dip in voltage when the ion touches the line. The current, unfortunately, is also somewhat dependent on the voltage used (Knoll, 1979). Phase one of this experiment was to find a plateau region where a relatively large change in voltage led to a relatively small change in output current.

The reactor was brought critical at 5 watts and then at 100 watts. Based on balancing a measurable output reading against the need to stay at low power, a power level of 50 watts was chosen for the experiment. The output of the CIC was run through an ammeter to a chart recorder. As the voltage was increased, the trace increased in magnitude, until it reached the plateau region at 630 volts on the positive chamber, 80 volts on the negative chamber.

Phase II. Measurement of Frequency Response of CIC

The compensated ion chamber, by design, is capable of measuring individual neutrons. This leads to a short dead time after measuring a neutron during which the detector is reduced in sensitivity. Because the CIC used for this experiment was a high capacitance device, it tended to blur these individual neutron peaks into a constant current signal at all but the lowest frequencies. This is desirable as the variable of concern is the average power level.

It was necessary to determine if the CIC would show a different magnitude of current for differing frequencies of neutron exposure at the same power level. If the ion chamber had a frequency dependence, it would have been necessary to construct a transfer function for it to distinguish between how the reactor operated on the input reactivity, and how the CIC operated on the reactor's output signal.

The first attempt to analyze the detector's frequency response was with a Frequency/Amplitude Response Monitor (FARM). This device was essentially a borated polyethylene shield for the CIC, with a rotating circular disc of reinforced cadmium, attached to a variable frequency motor, mounted on the front. The disc has 2 windows

in it to allow neutrons to pass, but the rest is an effective shutter. Figure 4 is a photograph of the back view of the FARM, figure 5 is a diagram of the front view.

By driving the motor at different frequencies, it was possible to expose the CIC to a time varying source of neutrons. Again, the trace was recorded on a chart recorder.

With this apparatus, the detector output is proportional to the total neutron fluence per window passing. This is an energy measurement, and hence is expected to be proportional not only to the neutron flux (power level) but also to the open shutter time. Open shutter time is inversely proportional to frequency, therefore, at a constant power level, Energy per pass should have a 1/f relationship to frequency.

Figure 6 is a graph of Energy vs Frequency. Note that it is close to a 1/f dependence, but there is significant variation. This is attributable to a frequency dependence in the ammeter and in the chart recorder. Both of these instruments are relatively slow to respond compared with the response time of the detector. To correct for this error, a second experiment was conducted. The current output of the detector was run across a resistor, and the resulting voltage signal was taken directly into a Memory Oscilloscope (Memoscope). The memoscope has no frequency dependence, and functions exactly like an oscilloscope, but has the additional capacity to freeze a "snapshot" of a trace for analysis.

Using the memoscope produced much more acceptable results. The peak energy of each window crossing was scaled by a factor of "f", this was to account for the 1/f dependence that was expected. Figure 7 is a graph of scaled peak energy vs frequency. It is linear, as expected, the slope corresponding to the area of the window in arbitrary units. If data were corrected for the area of the window, the energy output would be constant for all frequencies.

The fact that the compensated ion chamber had a constant current output for all frequencies means that the CIC has no frequency dependence. Therefore, it does not operate on the frequency content associated with the neutron density. Its measurements reflect the true power level of the reactor and can be used directly for noise analysis.

Phase 3 β / l Measurement

     The final phase of this experiment was to actually measure the frequency response of the TRIGA reactor. As has already been stated, the easiest way to measure this is to input a variety of frequencies, and measure the response of the reactor. Because the datum of interest is one specific frequency at which the frequency response changes, it was desirable to input as many frequencies as possible.

     One technique would have been to make a single frequency oscillator, and then to vary the frequency. This would have allowed data to be collected at a finite number of discreet point frequencies. A problem arises with this technique in that it is entirely possible to miss the desired frequency, by checking frequencies on either side of it, and never realize that it had been missed. It would have been difficult to know at exactly what frequency the device was oscillating.

     A much better technique would have been to steadily and continuously change the input frequency. This again has the problem of not knowing exactly which frequency is being input.   There is instrumentation which could have accurately determined the oscillation frequency, for example, a tachometer. It was decided that the benefits of such a system were not worth the costs of turning an expensive piece of gear into radioactive waste.

     The chosen method of inputing a reactivity perturbation into the reactor system was to input a string of random insert/remove signals, and then to Fourier Transform these to the frequency domain. The digital bit "1" was called insertion, and digital "0" was removal. Theoretically, an infinite string of random 1's and 0's would contain all possible frequencies as components. Therefore this random string would input the equivalent of white noise into the reactor, allowing a measurement of all possible frequencies.

     The problem with this method arises in that it was not possible to perform this experiment for an infinite amount of time. A finite run of random 1's and 0's develops certain unpredictable inconsistencies from white noise that would have been unacceptable. Therefore, rather than use a perfectly random series, a pseudo-random sequence was

employed.

A pseudo-random sequence is a string of 1's and 0's that, when repeated over time, approximate white noise. The best known, and most useful, class of pseudo-random sequences are those generated by digital shift registers. In this case, a 4 stage digital shift register with modula 2 adder feedback was used. This sequence is maximal at length 16, and very closely approximates white noise after a minimum number of repetitions. The time domain graph of this sequence is included as Figure 8, a frequency transform is shown as Figure 9. Note that this frequency transform does not show white noise as expected. This is primarily because of the system that was engineered to input the sequence into the reactor. This system took approximately 2 seconds per bit of the sequence, this requires an extremely long time to include the highest frequencies. This is compensated for in the analysis by looking at the actual input signal, not the ideal white noise. There are sufficient frequency components in the desired range (150 to 250 Hz) to accept this experiment as valid.

The next step was to insert this sequence into the reactor as a reactivity perturbation. The simplest method to perturb the reactor would have been to alternately insert and withdraw a control rod. This method was rejected for a number of reasons. Using a control rod to perturb the reactor would lead to a disturbance of one to two cents. This is too much for noise analysis purposes, as it would have changed the baseline power level of the core. As stated above, noise analysis relies upon a stationary signal, one that does not change its mean value with time. Use of a control rod would have also caused operator error to creep into calculations.

Another possible technique would have been to insert a piece of boron impregnated polyethylene slug into the Core Experimentation Tube (CET). By moving the slug vertically in the core, the boron would act as a poison of varying strength, depending upon its location in the core's vertically varying neutron flux. This technique was abandoned because of the design of the CET. Physically the CET is a long bent tube of aluminum. The bend is necessary to prevent radiation streaming up from the core through the air passageway of the tube. This bend also constricts the diameter of whatever is to be inserted into the tube. This bend was too narrow to permit associated

machinery for an oscillating slug.

The chosen solution was to input a change to the reactivity of the core by acting on the leakage neutrons. When the core is exposed to ER 2, approximately 120 degrees of its surface is separated from the air by only a thin layer of water and aluminum. This means that any neutrons that happen to exit the core in that direction will escape into the atmosphere of ER 2. By placing a piece of polyethylene near the core projection in the exposure room, neutrons that would have escaped from the core reflect from the polyethylene and return to the core. This is the equivalent of a few tenths of a cent worth of reactivity. It was not possible to quantify this because of the coarseness of the reactivity measuring instruments and procedures. The actual amount of reactivity inserted is not important to this experiment as long as it was small enough to ensure a stationary power level signal. Observations made over a long period of time of the reactor's power level when the oscillator was left in one configuration or the other indicated that there was in fact a small reactivity difference.

A sheet of cadmium, approximately 10 inch square placed between the core projection and the polyethylene block would absorb any neutrons that would have bounced back into the core, thus again denying these neutrons to the reactor system. This is now the equivalent of a few tenths of a cents of negative reactivity.

The sheet of cadmium is attached perpendicularly to a rotating disk, such that the plane of the cadmium is perpendicular to the plane of the disk. As the disk rotates 90 degrees, the cadmium sheet will go from "face on" to the core, where it will absorb neutrons, to "edge on" to the core, where it will allow the neutrons to pass by, bounce off of the polyethylene, and reflect back into the core. For the purposes of this experiment, "face on" is referred to as the "negative" position and "edge on" is referred to as the "positive" position with regards to reactivity. Figure 10 is a photograph of the Reactivity Insertion Oscillator (RIO) free standing inside a radiation work area. Figure 11 shows the RIO setup inside its shielding in ER 2, it is shown in the negative position.

To drive the oscillator in accordance with the pseudo-random sequence, coding was written in the form of a Turbo Pascal program. The first plan was to write a single program that would rotate the oscillator, and collect data using multiple channels of a

Digital Input / Output board. This plan was rejected simply due to the speed of the intended computer. For frequency transform, the highest frequency measurable is one half the sampling rate. This is the Nyquist frequency. With the computer both measuring data and controlling the oscillator, its rate of data gathering would be near 100 samples per second, well below the acceptable range.

This experiment required two independent computer systems. The first computer controlled the oscillator, while the second gathered data both on the power level from the core, and the position of the oscillator.

The command and control of the oscillator was from a 0 or 5 volt signal from the computer which latched a relay, latching a second relay. The second relay allowed motor current to flow, driving the oscillator in rotation until 1 of 2 magnetic switches mounted on the oscillator contacted. This interrupted the current latching the second relay, causing it to drop, disengaging the motor. Figure 12 is a diagram of the oscillator control system. Appendix 2 is the code for the control program.

The voltage across these magnetic switches was used to determine the position of the oscillator. The switch identified as switch #2 in the diagram was used for this position data. This is a normally closed magnetic switch. When the oscillator is in the positive position, switch #2 opens, switch #1 is closed, yielding a 5 volt potential across switch #2. When the oscillator starts to move, both switches close. The voltage drop across magnetic switch #2 is then due solely to line losses in the cable running from the control area into the exposure room. This voltage is close to 0.2 volts. Finally, when the oscillator is in the negative position, switch #2 is closed, #1 is open. This causes the voltage across switch #2 to drop to zero.

The second computer is capable of reading 1000 data points per second on four channels, simultaneously. The simultaneous data acquisition is necessary because the two signals must be analyzed against each other as a Cross Power Spectral Density. One of these channels is devoted to reading the voltage across magnetic switch #2, the other to the power level readings from the CIC.

The input reactivity into the system is given by the position of the oscillator. The other piece of data needed is the output of the core power level, or more specifically, the

noise portion of that signal. The ion chamber puts out approximately a 3 nA signal when exposed to a neutron flux in ER 2 at 50 watts. This signal was passed through an Action Pak, essentially an Op Amp which converts it into a 0 to 10 volt signal. The computer could have read this signal, but the important part is contained in the noise, which was a 0.1 volt oscillation at the extreme end of the scale. Due to precision problems with a 16 bit Digital input board, it is more desirable to have the important part of the signal be the full range of the voltage input. This signal was reverse biased to strip off the DC component, and then re-amplified so that the 0.1 volt noise became a 0 to 2.7 volt signal. Three volts was the maximum range of this particular channel of the data computer.

A chart recorder was used to get an initial look at the behavior of both the position readings of switch #2 and the signal from the ion chamber. It became readily apparent from the first test run that the cadmium sheet was acting as a shutter for the ion chamber. The neutron detector was offset from the RIO, but in the negative position, neutrons were bouncing from the polyethylene into the detector. A clear increase in neutron density was observed correlating to the positive position of the oscillator. This problem was solved by shielding the CIC in a block of borated polyethylene, with just its front end pointed at the core projection.

Another problem that arose was that the large inductive load of the motor relay latching was causing cross talk with other wires in the conduit. Specifically, the power signal was affected. A sharp spike in power was noted every time the relay closed, whether the motor was engaged or not. This was corrected by changing to a smaller relay. This partially corrected the problem. Eliminating a ground loop in a bad voltage source corrected the remainder of this problem.

A clean trace was then observed on the chart recorder, indicating that any major problems had been resolved. Three short test runs were made with the reactor most nearly critical. Energizing the oscillator device caused the reactor to become slightly sub-critical, but the point kinetics equations used to analyze the data are still accurate in the slightly sub-critical regime.

Several days later, a run was made in an attempt to gather a longer data set and to become more exactly critical. The reactor operations supervisor at the AFRRI reactor

modified one of the control rods so as to allow smaller corrections to criticality. This modification allowed perfect criticality to be maintained within a few milliwatts of average power for a period of over one hour. A 76 minute data run was then collected.

V. Data Analysis

The data that was gathered came out of the data computer as a integer type file, a continuous stream of numbers. Code was written to first extract the two channels of interest, and secondly to interpret the voltage signals of switch #2 as positional readings. Since the voltage read across the magnetic switch was constant when the cadmium sheet was in motion, it was necessary to interpolate to obtain a measure of the cross section of the sheet exposed to the core. For this interpretation process, a sinusoidal interpolation was used. This gives an excellent approximation for the actual face presented to the core by the cadmium.

Both of these codes operate on extremely large data sets, the data from a 1 minute run, represents 60000 numbers per channel. The data for the 76 minute run was 4.6 million numbers per channel. This required 36 megabytes of disk space, and involved physically shipping the computer to the U S Naval Academy for analysis.

These data sets took an extremely long time to process, even for the short runs. The code involved is included as Appendix 3. A reader utility was also written to verify the correct operation of the interpolator. The code allows the user to view selected data points, and ensure that spurious errors were not being introduced. This code is included as Appendix 4.

The data, after initial processing, was taken to the U S Naval Academy's CADIG computer facility, where it was processed by the MatLab software. With the input reactivity as the input channel and the power noise readings as the output channel, a transfer function was constructed for each test run.

The amount of data from the 76 minute run was too large for the computer facility to handle. For analysis purposes, it was necessary to limit the data set to the first 20 minutes of the experimental run.

Rather than use a Laplace transform, as is specified by the theory, transfer function estimates were obtained by using independent Fourier transforms of input and output data. This is allowed by Welch's method, but is inexact for a finite length signal. To correct for this inaccuracy, a Kaiser window filter was employed. A Kaiser window decreases the magnitude of the extraneous sidelobes that are introduced both by Welch's method and by the use of a finite length input sequence. (Uhrig, 1970) (Krauss, 1994)

VI. Discussion of Results

The estimated transfer function from the one minute run was not useful. One artifact shown might possibly be the expected corner frequency of β/l, but that is not clear from such a short run. The longer the run, the more frequencies are input to the system by the pseudo-random sequence. Over a one minute run, accurate results were not expected.

Figure 13 is a log plot of the magnitude of the Reactor Transfer function vs frequency plotted on a log scale. The data is a clouded scatter plot, shown is a third order curve fit. Third order was chosen because the theoretical plot has two inflection points and is therefore an odd order curve. Note that the breakpoint frequency of the plot corresponds to the higher corner frequency of the linear approximation of the theoretical transfer function. This breakpoint occurs at roughly 183 Hz.

The standard value of β for new TRIGA fuel is .007. The value of l is 39 E-6 seconds. This yields a predicted value of β / l of 179 Hz. It is expected that β / l will increase slightly over the lifetime of the core. 183 Hz is well within that expected range of increase.

A few factors were anticipated to contribute inaccuracies to the measurement. The first amongst these is the fact that, due to engineering constraints, approximately a 2.5 second transfer time was the best possible for the insertion rate. This means that higher frequencies were not directly present and were input over time by the pseudo random sequence. The fact that the run time was also limited to 20 minutes by the sheer amount of data denied the appearance of very low frequency components. The data

acquisition rate of 1000 samples per second effectively limited the highest frequency to 500 Hz, but even data in the 400 Hz to 500 Hz range is not as complete as it could be with a faster transition time and longer data sequence.

Uncorrelated, extraneous, white noise, introduced by random sources in the environment and amplified by the Operational Amplifiers, had the effect of adding a constant positive shift to the estimated transfer function. As this analysis looked only at the relative breakpoint, uncorrelated white noise did not interfere.

Had a specific frequency noise source, for example a 60 Hz source from the lighting and power systems, been introduced to the system, it would have been a specific point jump in the transfer function. This is the best explanation for the cloud effect on the data. The 3rd order fit effectively removes these spurious jumps in the transfer function by smoothing out the cloud. The large number of data points gathered preclude the usefulness of displaying them on the graph. The points are a large cloud, showing the desired trace as only a general trend.

VII. Concluding Remarks

As set forth in the theory, β / l, is expected to increase over the life of the core. Therefore, it is not unreasonable for β / l to have increased from 179 Hz to 183 Hz in the 30 years of operation of the AFRRI core. This is due simply because of the burn-up of the U-235 fuel, and is expected to continue. By tracking this β / l parameter over the next 15 years, AFRRI expects to develop a range at which it becomes necessary to refuel the core.

Independent measurements are now being planned at the AFRRI facility to determine β, this will allow l to be determined from this experimental ratio.
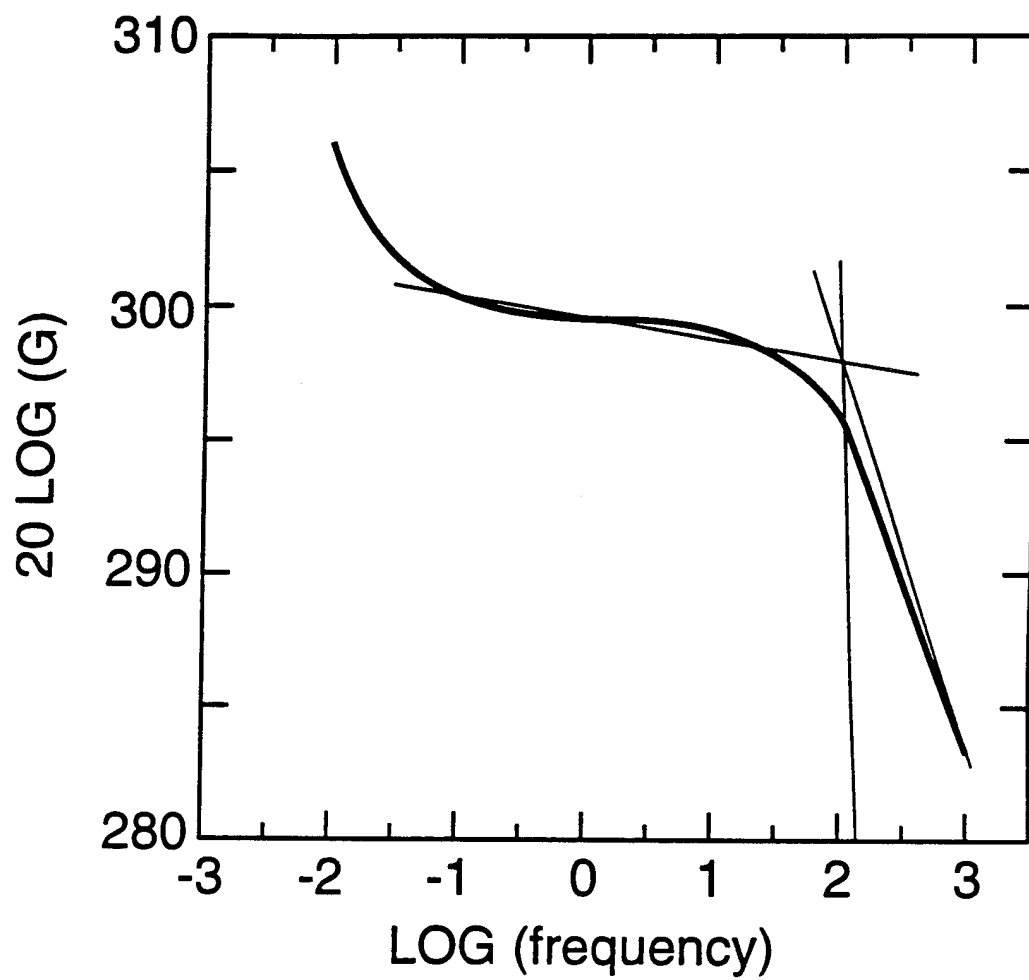
Figure #1: The Neutron Lifecycle
(Fort Belvoir text)

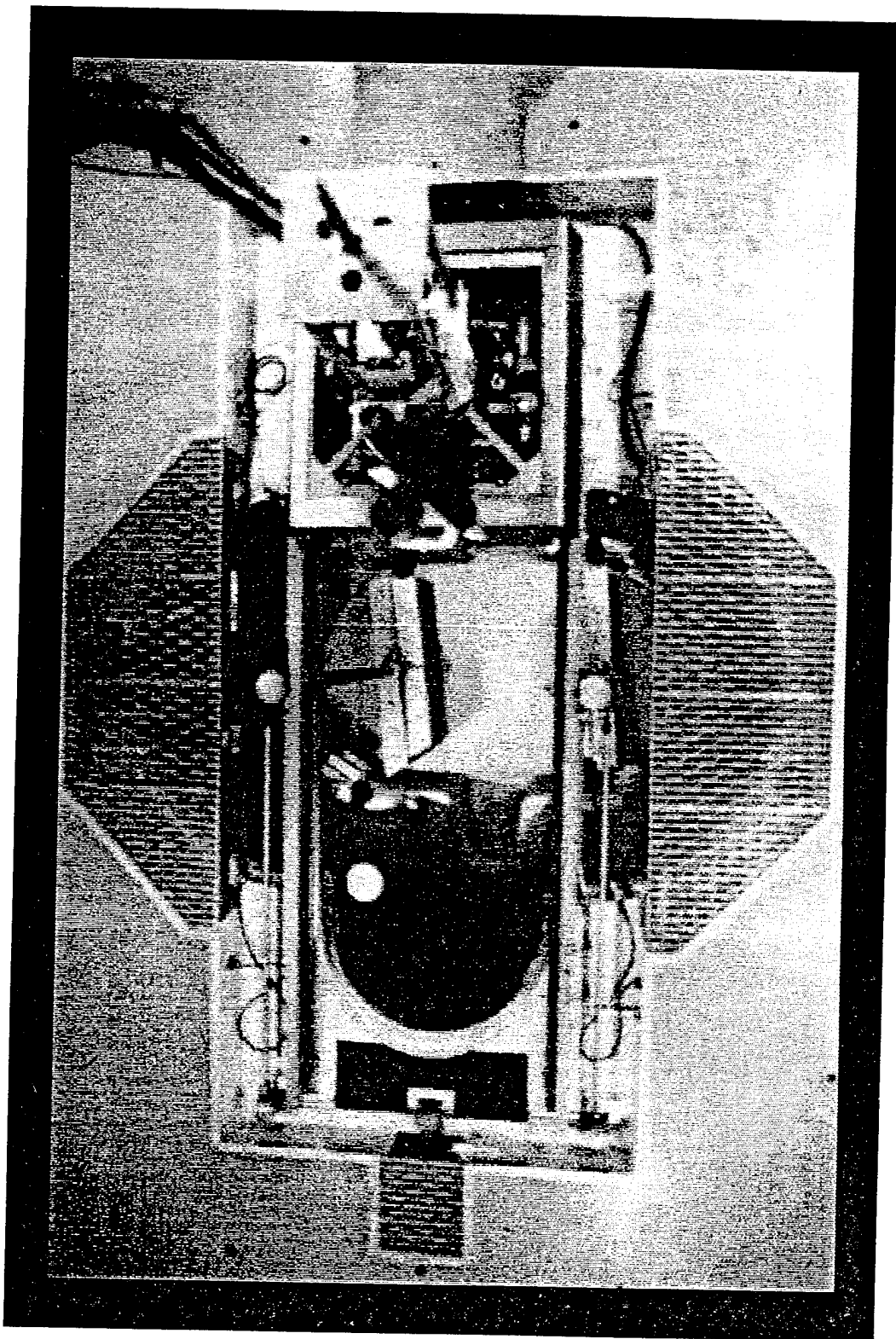Figure #2: Theoretical Plot of Transfer Function

Both Axes Dimensionless

Figure #3: Overhead View of Pool and Core

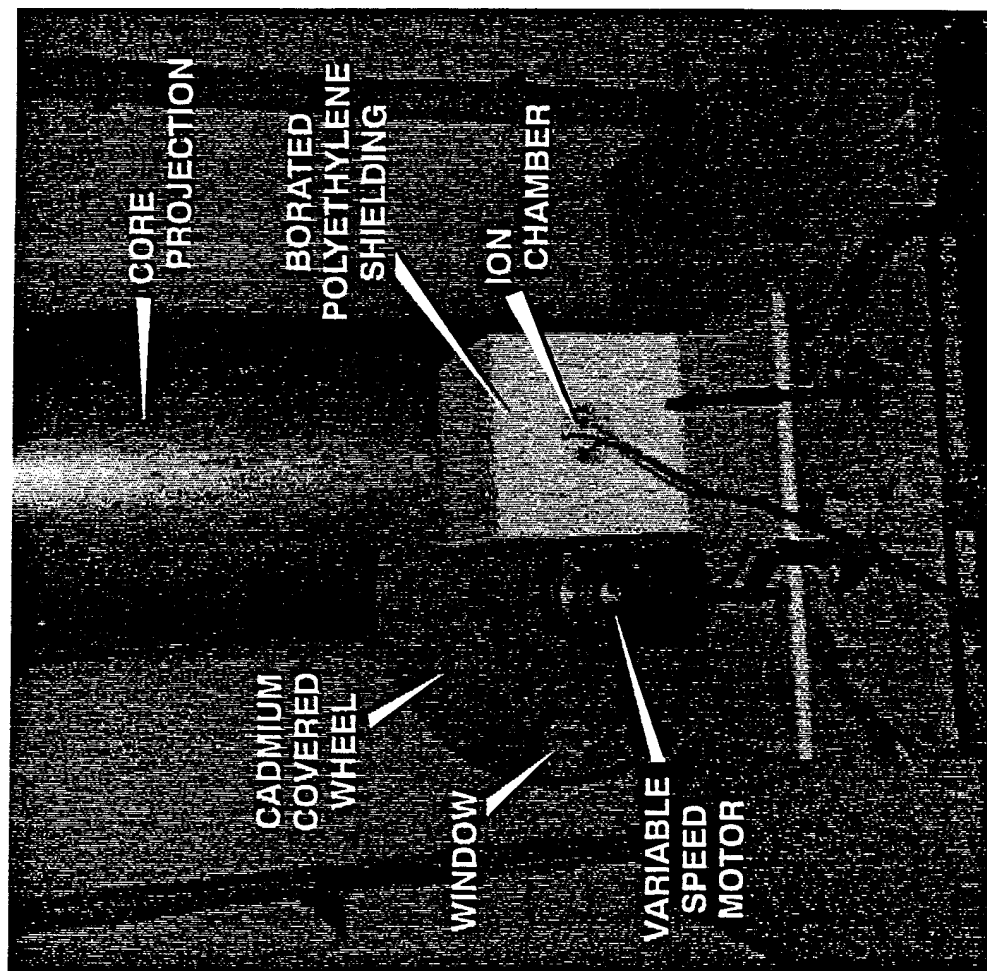Figure #4: Frequency Amplitude Response Monitor

Frequency Amplitude Response Monitor

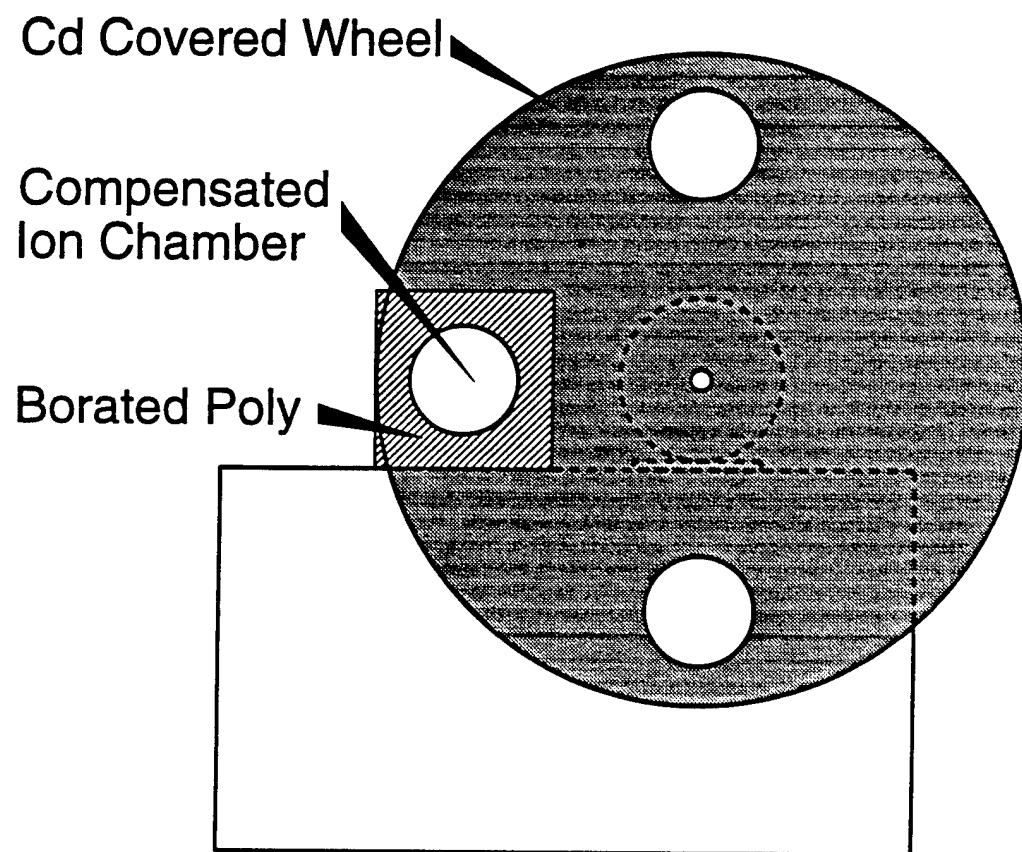Cd Covered Wheel

Compensated
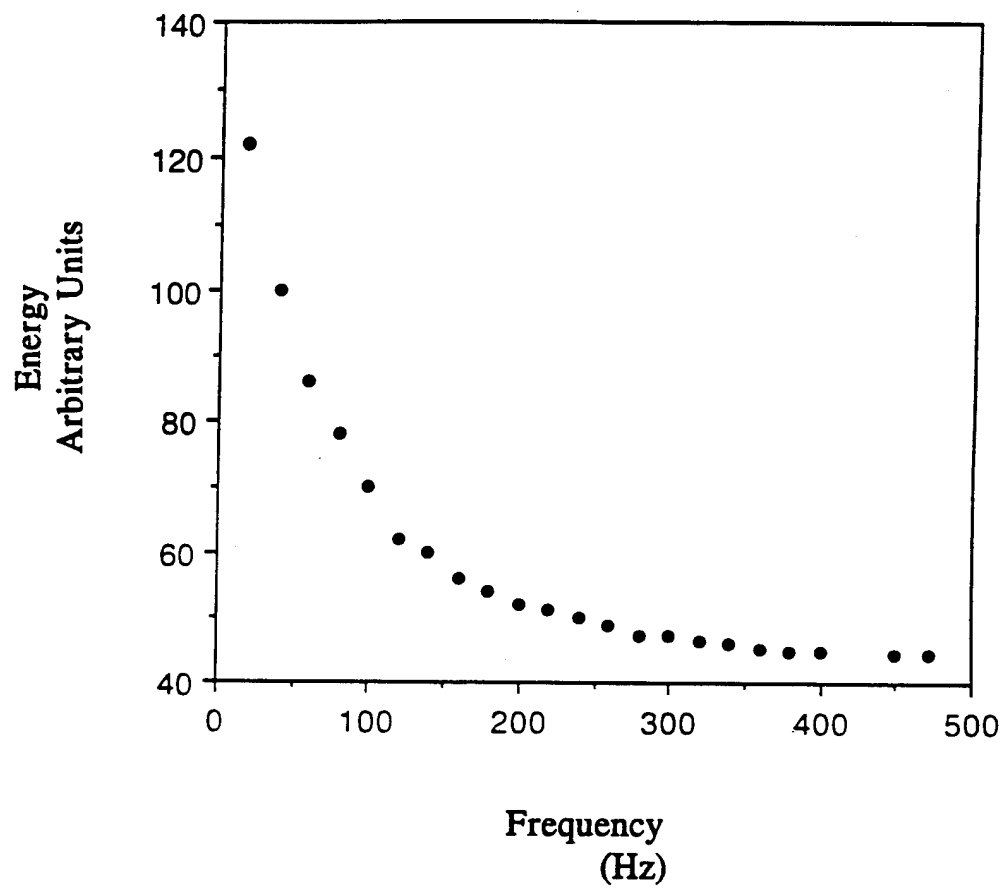Ion Chamber

Borated Poly

Figure #5: Frequency Amplitude Response Monitor

Figure #6: Frequency Dependence of CIC
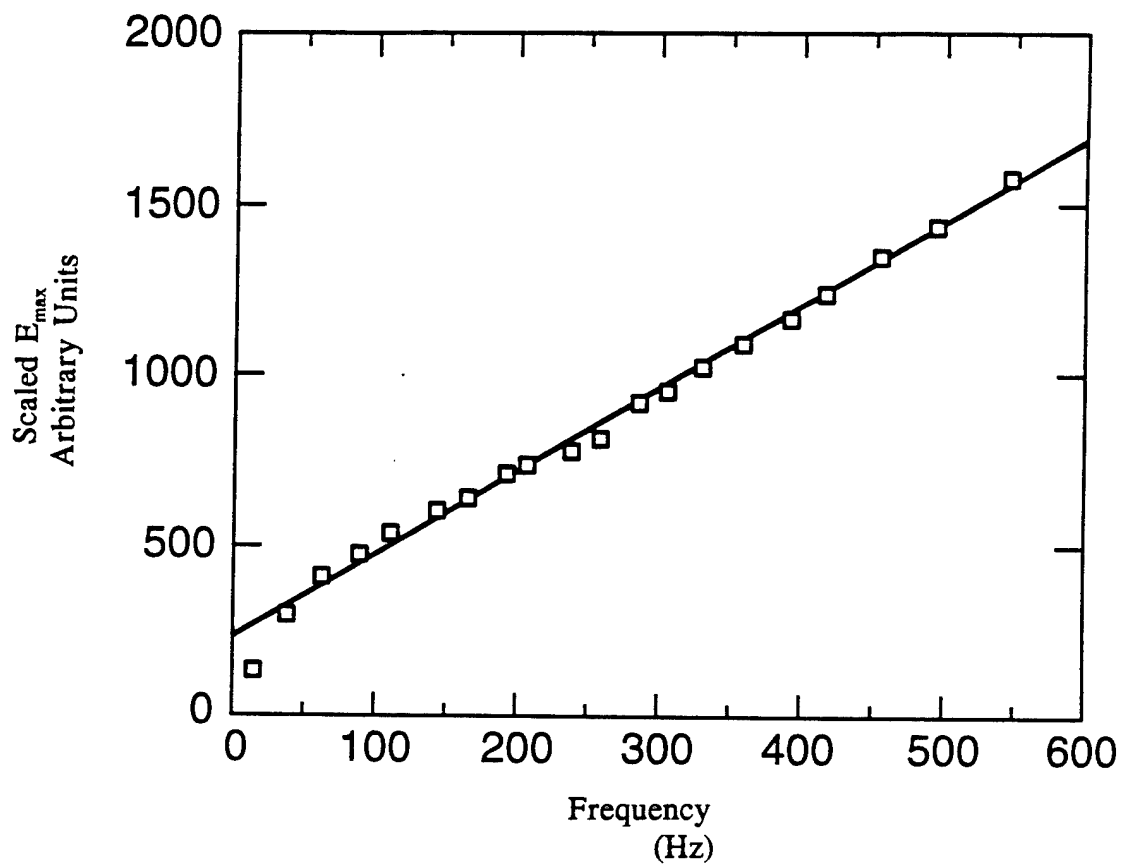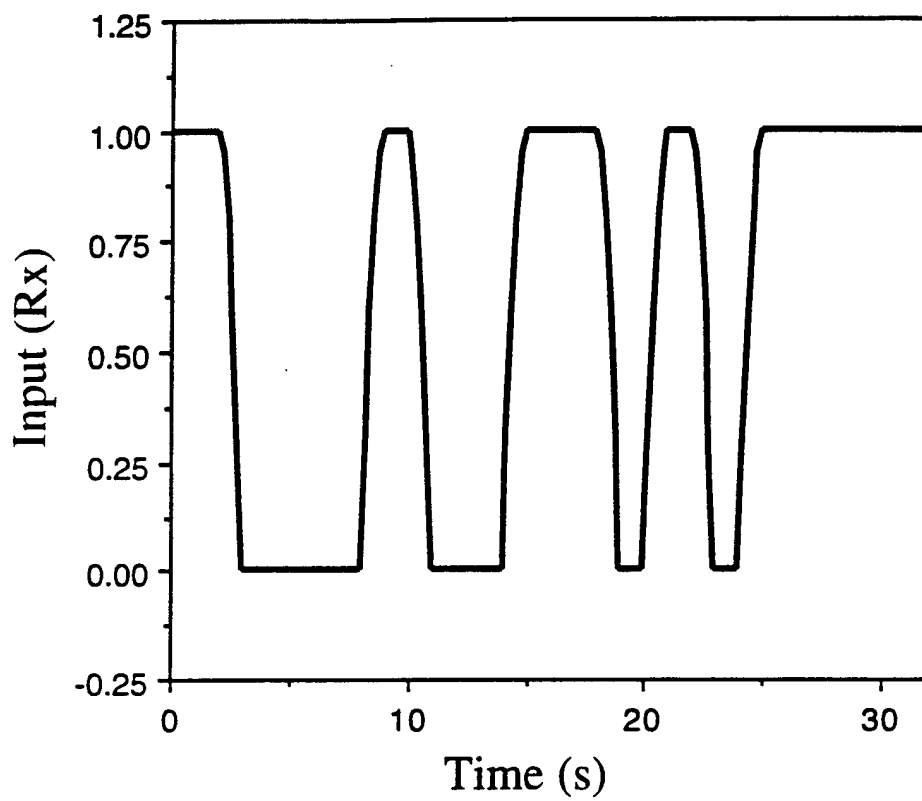
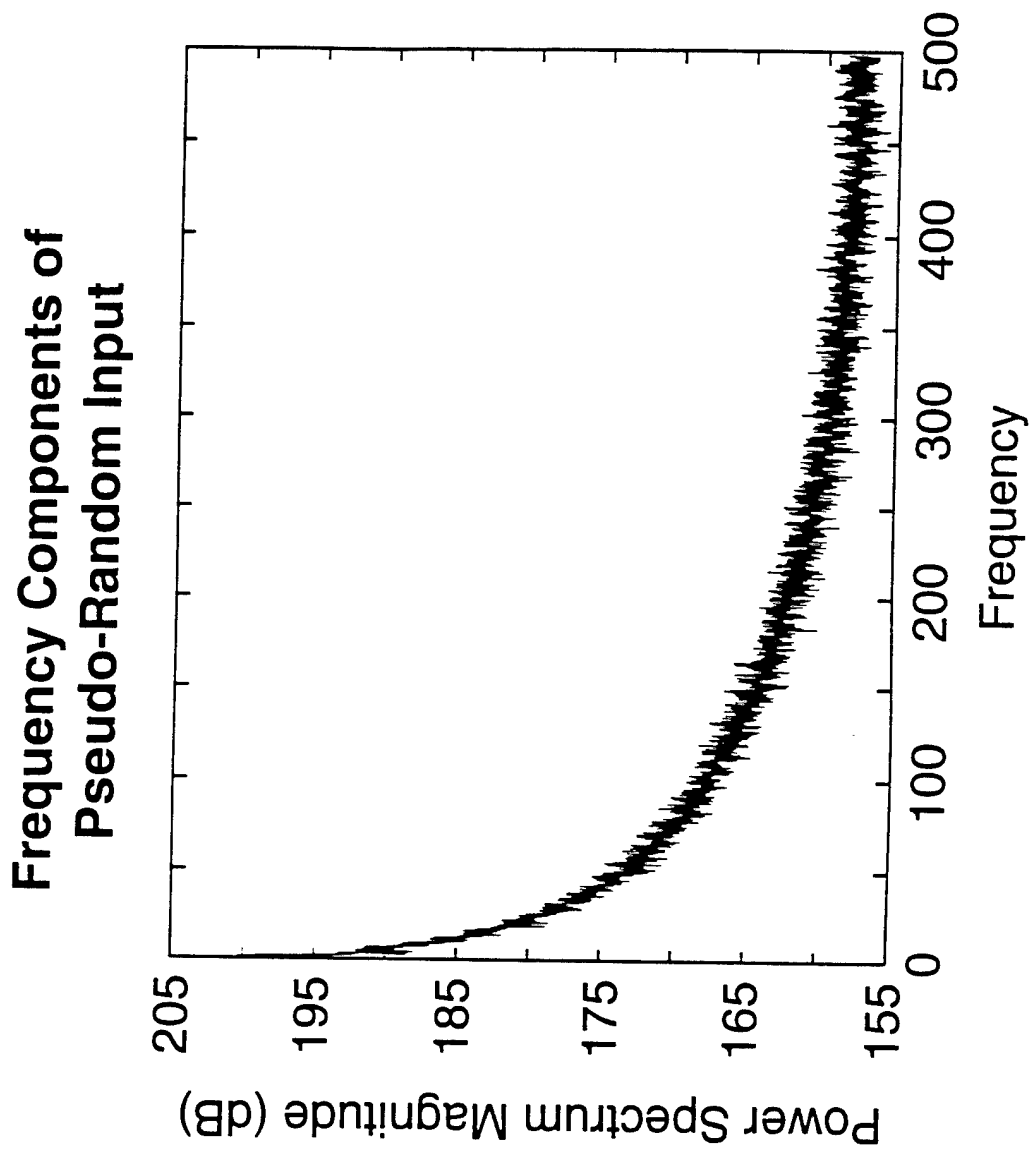Figure #7: Frequency Dependence of CIC

Figure #8: Input Reactivity Perturbation

Figure #9: Frequency Components of Perturbation
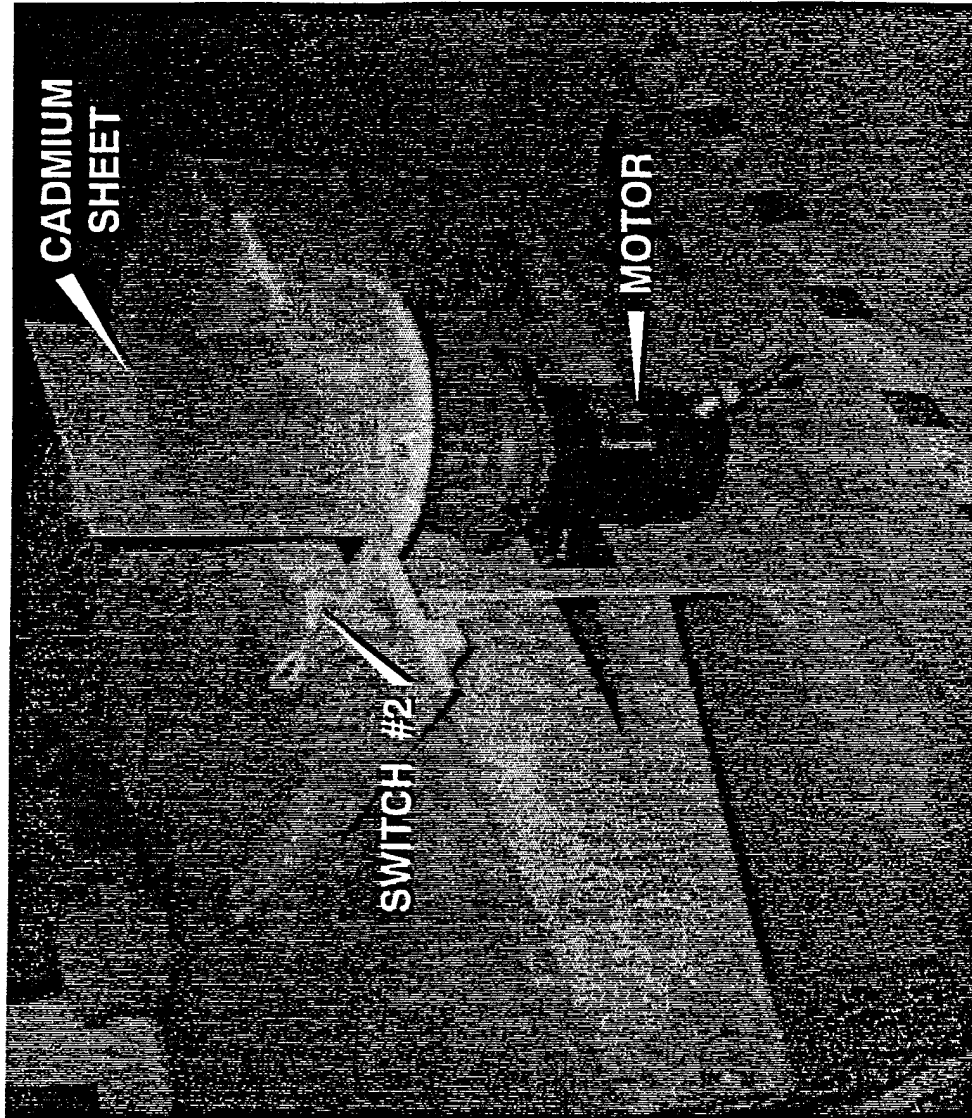
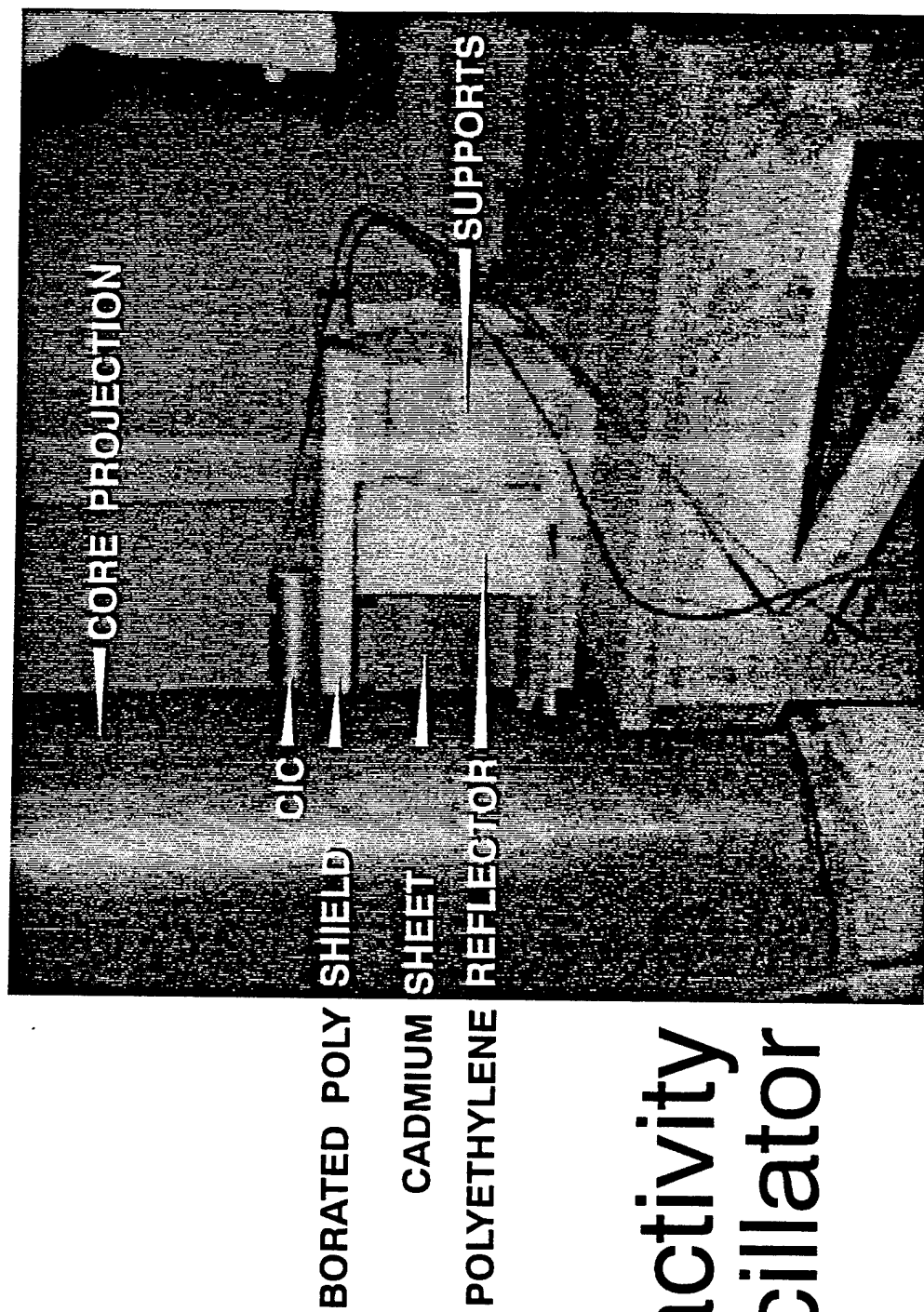# Reactivity Oscillator

Figure #10: Reactivity Oscillator

Figure #11: Reactivity Oscillator

Data Acquistion System
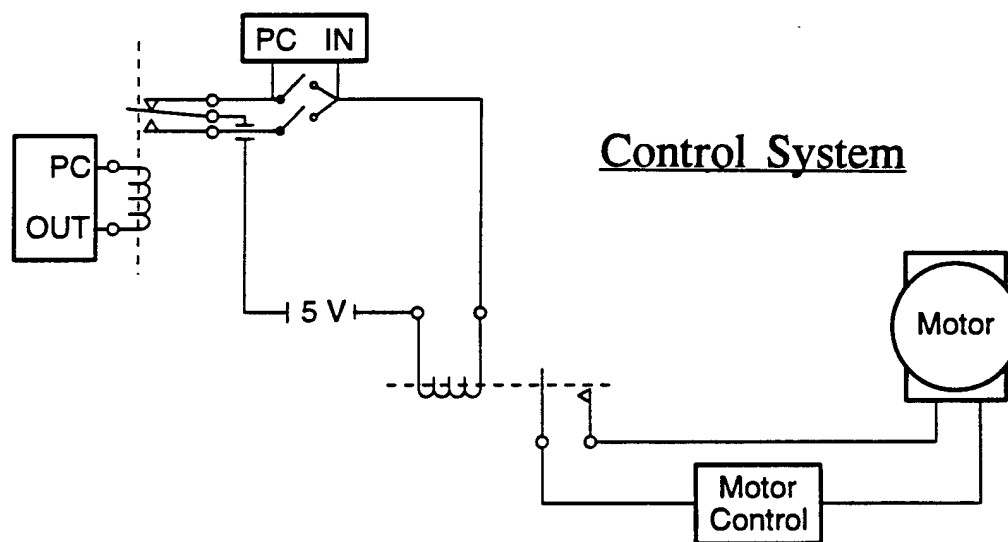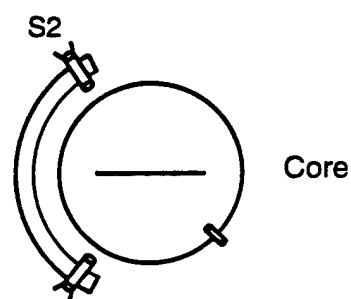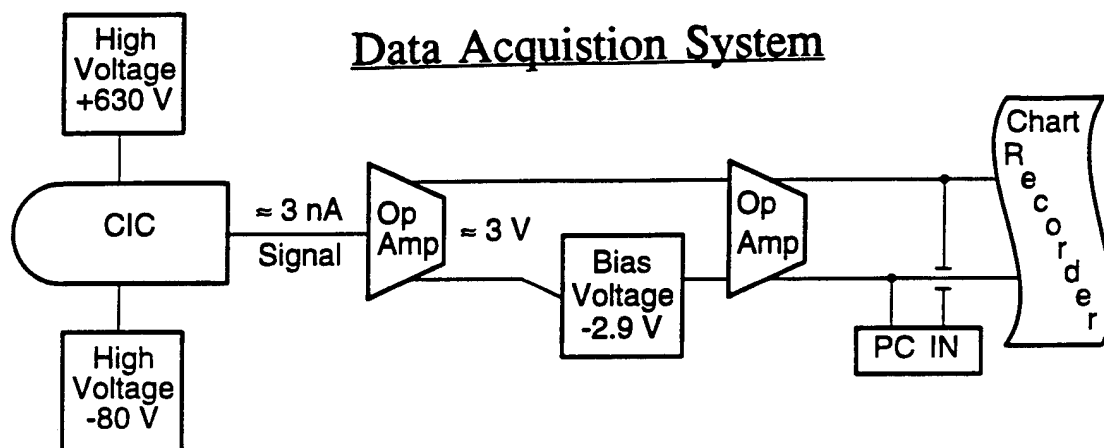

Control System
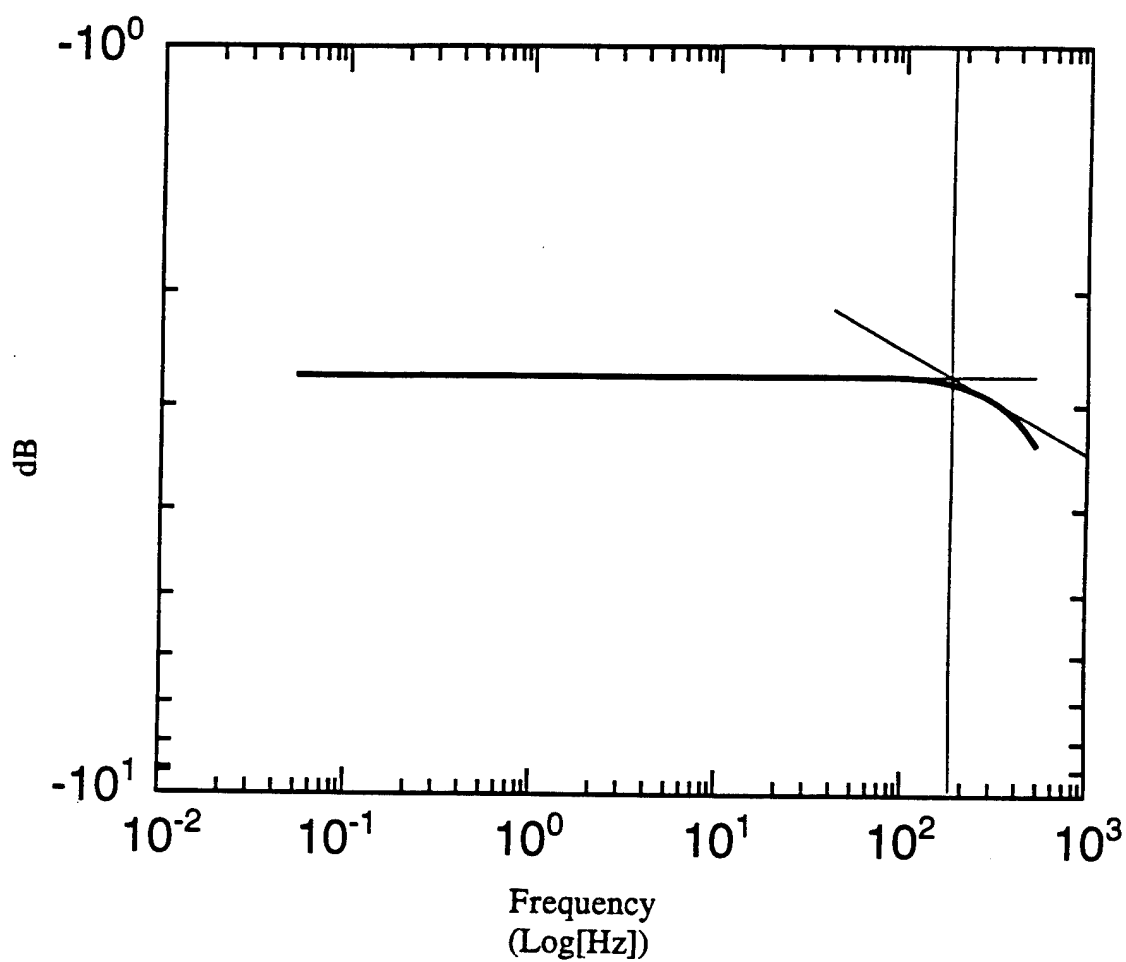
Figure #12: Block Diagrams

Figure #13: Reactor Transfer Function

Reactor Noise Analysis
Bibliography

1. Hetrick, D L, <u>Dynamics of Nuclear Reactors</u>, University of Chicago Press, USA, 1971.

2. Kerlin, T W, <u>Frequency Response Testing in Nuclear Reactors</u>, Academic Press, New York, 1974.

3. Knoll, G F, <u>Radiation Detection and Measurement</u>, Wiley & Sons, USA, 1979.

4. Krauss, TP, et al., <u>Signal Processing Toolbox User's Guide</u>, The Mathworks Inc., Massachusetts, 1994.

5. Thie, J A, <u>Power Reactor Noise</u>, American Nuclear Society, Illinois, 1981.

6. Uhrig, R E, <u>Random Noise Techniques in Nuclear Reactor Systems</u>, Ronald Press, New York, 1970.

7. US Atomic Energy Commission, <u>Fundamentals of Nuclear Reactor Engineering</u>, Fort Belvoir, 1971.

Papers

1. Moore, M, "The Determination of Reactor Transfer Functions from Measurements at Steady Operation", Nuclear Science and Engineering, 3.387-394, 1958.

2. Bars, B, "The Delayed Neutron Fraction-Neutron Lifetime Ratio in a TRIGA reactor from Noise Analysis", Nuclear Science and Engineering, 30.104-108, 1967.

3. Warren, L D, et al., "The Study of Nuclear Power System Dynamics Using a Pseudo-Random Binary Input," Annals of Nuclear Science and Engineering, 1.49-53, 1974.

4. Porto, J, "Analysis of Dynamic Perturbations in the RA-2 Zero Power Reactor," Annals of Nuclear Science and Engineering, 1.567-572, 1974.

Derivation of Reactor Transfer Function

$$\frac{\partial n}{\partial t} = \frac{\rho - \beta}{l} + \sum \lambda_i C_i + q$$

Point

Kinetics

Equations

$$\frac{\partial C_i}{\partial t} = \frac{\beta_i}{l} n - \lambda_i C_i$$

Eq 1a

Eq 1b

Assumptions:

$$n = n_o + \delta n$$

$$C_i = C_{i_o} + \delta C_i$$

$$\rho = \rho_o + \delta \rho$$

Original Value

plus

Small Oscillations

Eq 2a

Eq 2b

Eq 2c

$$q = q_o$$

Constant

Eq 2d

$$C_{i_o} = \frac{\beta_i n_o}{\lambda_i l}$$

$$q_o = -\frac{\rho_o n_o}{l}$$

Equilibrium States

from

Point Kinetics Equations

Eq 2e

Eq 2f

Where variables are defined:

n = neutron density
C_i = Precursor Density
t = time
ρ = reactivity
β = delayed neutron fraction
l = neutron generation time
λ_i = Precursor decay constant
q = source strength

Substituting Equations 2 into Equations 1

$$\frac{\partial}{\partial t}\,\delta n = \frac{\rho_o n_o}{l}\,\delta n + \sum \lambda_i \delta C_i + \frac{n_o}{l}\,\delta p + \frac{1}{l}\,\delta \rho \,\delta n \qquad \text{Eq 3a}$$

$$\frac{\partial}{\partial t}\,\delta C_i = \frac{\beta_i}{l}\,\delta n - \lambda_i \delta C_i \qquad\qquad \text{Eq 3b}$$

Assume $|\delta n| \ll n_o \rightarrow \delta \rho \delta n$ *is negligible*   Eq 4

Small Oscillation allows Laplace Transforms:

$$\delta N(s) \;=\; \int \delta n(t) e^{-st} dt \qquad output \qquad\qquad \text{Eq 5a}$$

$$\delta \Gamma_i(s) \;=\; \int \delta C_i(t) e^{-st} dt \qquad\qquad\qquad \text{Eq 5b}$$

$$\delta R(s) \;=\; \int \delta \rho(t) e^{-st} dt \qquad input \qquad\qquad \text{Eq 5c}$$

The following Laplace transform is from
Hetrick-Dynamics of Nuclear Engineering:

$$\delta N(s) = \frac{n_o \delta R(s)}{ls + \beta - \rho_o - \sum \dfrac{\beta_i \lambda_i}{s + \lambda_i}} \qquad\qquad \text{Eq 6}$$

Transfer Function:

$$G(s) = \frac{OUTPUT}{INPUT} = \frac{\delta N(s)}{\delta R(s)} = \frac{n_o}{ls + \beta - \rho_o - \sum \dfrac{\beta_i \lambda_i}{s + \lambda_i}} \qquad \text{Eq 7a}$$

$$G(s) = \frac{n_o}{[l + \sum \frac{\beta_i}{s + \lambda_i}] s - \rho_o}$$

Eq 7b

For a single group of Delayed neutrons:

$$G(s) = \frac{n_o}{[l + \frac{\beta}{s + \lambda}] s - \rho_o}$$

Eq 8a

$$= \frac{n_o(s + \lambda)}{ls^2 + (\beta + \lambda l - \rho_o) s - \lambda \rho_o}$$

Eq 8b

For a Critical Reactor, $\quad \rho_o = 0$

$$G(s) = \frac{n_o(s + \lambda)}{ls^2 + (\beta + \lambda l) s} = \frac{n_o(s + \lambda)}{ls[s + \lambda + \beta/l]}$$

Eq 9

$\lambda << \beta / l$

$$G(s) = \frac{n_o(s + \lambda)}{(ls)(s + \beta/l)}$$

Eq 10

In Bode Form:

$$G(s) = \frac{n_o}{\beta \lambda} * \frac{(\frac{s}{\lambda} + 1)}{s(\frac{l}{\beta} s + 1)}$$

Eq 11

Appendix #2

```
PROGRAM  RIOCONTROL;
{by MIDN J C Ventura}

USES  CRT, DOS, tur_sub, dtst_tls, dtst_err, dtst_xmm, WINDOWS;

Const
   RSEQ : Array[0..15] of integer= (1,0,0,0,1,0,0,1,1,0,1,0,1,1,1,1);
   driver_name = 'DT2812$0'#0;  {driver's log. name nul ter}
   a_unit = 0;

Var
(*board prog variables*)
   digitalio    : DIGITALIO_STRUCT;
   sgl          : SINGLE_STRUCT;
   i            : integer;
   istat    :    integer;
   handle   :    integer;
   xbsize   :    integer;
   adsetup  :    WORD;
   dasetup  :    WORD;
   d_name   :    string[15];

(*this program's variables*)
   RCOUNT      :integer;              (*0..15, where are we in PRS*)
   OPOS        :Longint;              (*ordered position*)
   dopos: integer;
   HOUR, MINUTE, SECOND,sec100  : WORD;
   HOURR, MINUTER, SECONDR,sec100r  : REAL;
   HOURS, MINUTES, SECONDS,sec100s  : STRING;
   NHOUR, NMIN,NSEC: REAL;
   EHOUR, EMIN,ESEC: REAL;
   buffer:string;
   TIME        :real;                (*ACTUAL # OF SECONDS*)
   ERR: integer;
   nextpostime:real;
   Poshold:real;
   ch: char;
   endtime: real;
   runtime:real;

Label
```

```
      overhere;
   Label
      loopback;

   Procedure TIMECHECK;
   BEGIN;
      GETTIME(HOUR,MINUTE,SECOND,SEC100);
      STR(HOUR,HOURS);
      STR(MINUTE,MINUTES);
      STR(SECOND,SECONDS);
      STR(SEC100,SEC100S);
      VAL(HOURS,HOURR,ERR);
      VAL(MINUTES,MINUTER,ERR);
      VAL(SECONDS,SECONDR,ERR);
      VAL(SEC100S,SEC100R,ERR);
      TIME:=((HOURR*3600)+(MINUTER*60)+(SECONDR)+(SEC100R/100));
   END;

   PROCEDURE CYANBLACK;
   BEGIN;
      TEXTCOLOR(cyan);
      TEXTBACKGROUND(black);
   END;

   PROCEDURE BLACKCYAN;
   BEGIN;
      TEXTCOLOR(BLACK);
      TEXTBACKGROUND(CYAN);
   END;

   Procedure WHONEXT;
   Begin;
      NEXTPOSTIME:= TIME + POSHOLD;
      RCOUNT:=RCOUNT+1;
      RCOUNT:=(RCOUNT MOD 16);
      OPOS:=RSEQ[RCOUNT];

      if opos=0 then opos:=$00;
      if opos=1 then opos:=$FF;
      digitalio.dio_value:=opos;
      istat := dt_set_dio(a_unit, handle, digitalio);
   End;

   (* MAIN program STARTS here*)
```

```
BEGIN;
 OVERHERE:
  CLRSCR;
  RCOUNT:=-1;
  SETGROWDELAY(60);
  SETSHADOWOFF;
  SETTITLECENTER;
  BLACKCYAN;

  GROWWINDOW(5,3,77,15,'Reactivity Insertion Oscillator - by MIDN JOHN
C VENTURA');
  WRITELN('          ');
  WRITE('How many seconds between Position Decisions?  <CR>=2 (1-10) :
');

  READLN(BUFFER);
  VAL(BUFFER,POSHOLD,ERR);
  IF ERR<>0 THEN
  begin;
    POSHOLD :=2;
    gotoxy(66,2);
    writeln(poshold:4:2);
  end;
  poshold:=poshold - (0.04);  (*SCALER headstart*)

  WRITE('How long should I run for?  <CR>=1 min (minutes) : ');

  READLN(BUFFER);
  VAL(BUFFER,runtime,ERR);
  IF ERR<>0 THEN
  begin;
    runtime:=1;
    gotoxy(66,3);
    writeln(runtime:4:2);
  end;

  WRITE('       ARE THESE VALUES OK?  ');
  CH:=READKEY;
  IF UPCASE(CH) = 'N' THEN goto OVERHERE;

  (*initialize and reset dt2812*)
    d_name :=driver_name;
    istat:= dt_initialize (d_name[1], handle);
    istat:= dt_reset (a_unit, handle);
```

```
(*setup op parameters of dio port and dac channel*)
  sgl.sing_dacs := DAC_0;
  digitalio.dioport := PORT_1;
  digitalio.command := XFER_VAL;
  digitalio.direction :=DIO_OUTPUT;

(*WE HAVE USER INPUTS, NOW run Oscillator*)
gotoxy(2,3);
clrscr;
setgrowdelay(100);
setshadowoff;
settitlecenter;
cyanblack;
growwindow(5,3,77,15,'DATA COLLECTION MODE');
Writeln('                    SAY WHEN');
CH:=readkey;

timecheck;
endtime:=time + (runtime * 60);


 ehour:=trunc(endtime/3600);
 emin:=trunc((endtime-ehour*3600)/60);
 esec:=endtime-ehour*3600-emin*60;


clrscr;
writeln(' ');
Writeln('I am now a: Moving the Reactivity Rotator- according to the PRS');
Writeln('The other computer should be');
Writeln('           a: gathering power data from the CIC');
writeln('           b: gathering magnetic switch data to derive position');
writeln('');
writeln('');
writeln('');
writeln('ordered position      time        nextchange time      endtime');

 begin;
   loopback:
   timecheck;
   if time <= nextpostime then goto loopback;
   whonext;
   gotoxy(5,10);

   dopos:=Trunc(opos/255);
```

```
  nhour:=trunc(nextpostime/3600);
  nmin:=trunc((nextpostime-nhour*3600)/60);
  nsec:=nextpostime-nhour*3600-nmin*60;


  write('  ',dopos:1);
  Write('          ',hours,':',minutes:2,':',seconds:2,'.',sec100s:2);
  Write('        ',nhour:2:0,':',nmin:2:0,':',nsec:5:2);
  Writeln('        ', ehour:2:0,':',emin:2:0,':',esec:5:2);


  if time < endtime then goto loopback;
end;


(*END ROUTINE*)
clrscr;
setgrowdelay(60);
setshadowoff;
settitlecenter;


growwindow(5,3,77,15,'DONE WITH DATA COLLECTION');
writeln(' ');
writeln('We are done with this run.');
writeln('');
writeln('Shall we do another run?');
ch:= readkey;
if upcase(ch) = 'Y' then goto overhere;
(*terminate communications*)
OPOS:=2050;
istat :=dt_single_acq(a_unit, handle, DA_SECTION, OPOS, sgl);
istat := dt_terminate(handle);
writeln ('terminating   = ',istat);
end.
```

```
program intread;
{by MIDN J C Ventura}

uses crt,windows;

label  separate;
label  interpolate;
label  mainloop;
label  outtahere;

type   stack = array[0..99] of integer;

Var
  myfile :string[12];
  filebuff :string[12];
  stackfile :file of integer;
  interfile :file of integer;
  outfile: file of integer;
  realfile:file of integer;    .
  low, high,transit, killit, done : boolean;
  ch0,ch1,ch2,ch3 :stack ;
  c,l,che,h0,h1,h2,h3 :integer;
  r0,r1,r2: integer;   (*not really reals*)
  theta : real;
  e,i,j,k : longint;
  FS,where, fsold,fnum :longint;
  check:longint;
  see,ch : char;
  option:integer;


Procedure Blackcyan;
  begin;    textcolor(black);    textbackground(cyan);   end;
Procedure redblack;
  begin;    textcolor(red);    textbackground(black);  end;
Procedure Cyanblack;
  begin;    textcolor(cyan);    textbackground(black);  end;

Procedure Geth0;
 begin;
  read(interfile,h0);
```

```
   if h0=-32768 then h0:=-32765;
  h0:=abs(h0);
 end;


Procedure Checkhigh;
 begin;
   check:=h0;
   for che:=1 to 4 do
    begin;
      geth0;
      check:=check+h0;
    end;
   check:=check div 5;
   if check>15000 then
     begin;
       high:=true;
       low:=false;
       transit:=false;
     end;
    where:=filepos(interfile);
    where:=where-4;
    seek(interfile,where);
   end;


Procedure Checklow;
 begin;
   check:=h0;
   for che:=1 to 4 do
    begin;
      geth0;
      check:=check+h0;
    end;
   check:=check div 5;
   if check<100 then
     begin;
       high:=false;
       low:=true;
       transit:=false;
     end;
    where:=filepos(interfile);
    where:=where-4;
    seek(interfile,where);
   end;
```

```
Procedure Firsttrans;
 Begin;
 (*If 1st point is in transit, this routine detects where it is going*)
 reset(interfile);
 l:=-1;
 while done=false do
   begin;
     l:=l+1;
     seek(interfile,l);
     geth0;
     if h0<100 then checklow;
      if low=true then
       begin;
         h1:=1000;
         done:=true;
       end;
     if h0>20000 then checkhigh;
      if high=true then
        begin;
         h1:=0;
         done:=true;
        end;
   end;

 end;


Procedure Checktrans;
 begin;
  (*This routine checks to make sure that we really are in transit
     and not just reading a glitch*)
     check:=h0;
     for che:=1 to 4 do
       begin;
        geth0;
        check:= check + h0;
       end;
     check:=check div 5;
     if (check<15000) and (check> 100) then
       begin;
        transit:=true;
        high:=false;
        low:=false;
       end
     else transit:=false;
```

```
              (*100 < avg value of 5 points < 15000*)
      where:=filepos(interfile);
      where:=where-4;
      seek(interfile,where);

  end;

Procedure Lowinterpolate;
  begin;
  (*This routine gets called when we are sure of a transit*)
  if e>0 then
    begin;
     close(interfile);
     reset(outfile);        (*This first segment dumps the buffer*)
     seek(outfile,i);
     for c:=0 to e-1 do
      begin;
       h2:=ch0[c];
       write(outfile,h2)
      end;
     close(outfile);
     i:=i+e;   (* e counts from 0 to 99, but the eth point is bad*)

    end;
  (*Starting here, we count how many 'trans' points we have*)
  c:=0;

  if i>= fs-105 then killit:=true;

  if h1>950 then
     begin; (*This means we are going high to zero*)
      reset(interfile);
      seek(interfile,i+e);
      while low=false  do  (*NOT ZERO YET*)

       begin;
         c:=c+1;
         geth0;
         checklow;
       if (i+e+c) > (fs-100) then
        begin;
        low:=true;   (*Makes fileend look like a true*)
        high:=false;
        transit:=false;
```

```
            killit:=true;
          end;
        end;
     close(interfile);
     c:=c-1;

       (*We now have spent c counts in transit, do 90 deg. down sinusoid*)
     reset(outfile);
       seek(outfile,i);
       for l:=1 to c do
         begin;
           h1:=Trunc(1000*cos(1.5708*(1/c)*l));
           write(outfile,h1);
         end;
       i:=i+c+e;
       close(outfile);
       transit:=false;   (*This is to make sure that we don't enter next loop*)
     end;


   if (h1<50) and (transit=true) then
     begin; (*This means we are going zero to high*)
       reset(interfile);
       seek(interfile,i+e);

       while (high=false)  do  (*NOT HIGH YET*)

         begin;
           c:=c+1;
           geth0;
           checkhigh;

         if (i+e+c) > (fs-100) then
           begin;
             high:=true;   (*Makes fileend look like a true*)
             low:=false;
             transit:=false;
             killit:=true;
           end;
         end;
       close(interfile);
       c:=c-1;
       (*We now have spent c counts in transit, do 90 deg. up sinusoid*)
       reset(outfile);
```

```
    seek(outfile,i);
    for l:=1 to c do
      begin;
       h1:=Trunc(1000*sin(1.5708*(1/c)*l));
        write(outfile,h1);
      end;
     i:=i+e+c;
     close(outfile);
    end;


  (* This IS COMMON TO HIGH->LOW and LOW->HIGH *)
  e:=99; (*end this loop for e's*)
  transit:=true;   (*next loop in main prog checks for transit,
                 we don't want it to run this routine and close interfile,
                 then run that next one, and close it again*)
   end;



(*MAIN PROGRAM*)
begin;
 (*disclaimer screen*)
  clrscr;
  setgrowdelay(60);
  setshadowoff;
  settitlecenter;
  cyanblack;
  push;

  Growwindow(5,3,77,20,'Initial Data Manipulator - by MIDN JOHN C VENTURA,
USN');
   redblack;
   Writeln('                  A FEW MINOR DISCLAIMERS');
   writeln('');
   writeln('1. This program will quickly analyse a large amount of data.');
   writeln('    The catch is that it operates in blocks of 100 data points.');
   writeln('');
   writeln('2. This means that if you have 720224 data points,');
   writeln('    ie. slightly over a 3 minute run, you will lose the last');
   writeln('    56 points from each channel.');
   writeln('');
   writeln('3. If you lose the max amount of data, ie 99 points, ');
   writeln('    this is equivalent to losing 99 milliseconds of data.');
   writeln('');
   writeln('4. In the interest of saving space, the program deals with ');
```

```
writeln('    integers, from 0 to 20000 from the power channel,');
writeln('    and from 0 to 1000 after position interpolation.');
writeln('');
writeln('5. Integers take up 2 bytes each of disk space.');
writeln('');
cyanblack;
writeln('                        Any Key to Continue');

ch:=readkey;
pop;


mainloop:
(*initial selection screen*)
clrscr;
setgrowdelay(60);
setshadowoff;
settitlecenter;
blackcyan;

Growwindow(5,3,77,15,'Initial Data Manipulator - by MIDN JOHN C VENTURA,
USN');
Writeln('');
Writeln('Which Option would you like:  ');
Writeln('                        1) Separate data into power and position');
Writeln('                        2) Interpolate position data');
Writeln('                        3) END');
writeln('    ');
write('                choice:  ');
readln(option);
if option=1 then goto separate;
if option=2 then goto interpolate;
if option=3 then goto outtahere;
goto mainloop;

(*Option #1: SEPARATE*)
separate:
clrscr;
setgrowdelay(60);
setshadowoff;
settitlecenter;
cyanblack;

Growwindow(5,3,77,15,'Data Separator');
Writeln('');
```

```
(*setup save files*)
 assign(outfile,'power.dat');
 rewrite(outfile);
 assign(outfile,'posit.dat');
 rewrite(outfile);
(*done setting up save files*)

write('What file would you like to read? ');
readln(myfile);
assign(stackfile,myfile);
reset(stackfile);
fs:=filesize(stackfile);
fnum:=trunc((fs/4)/100);
writeln('This file has ',(fs/4):8:0,' data points per channel');
writeln('Do you wish to see the data as I extract it? ');
readln(see);

for i:=0 to (fnum-1) do
  begin;
    assign(stackfile,myfile);          (*READ points IN*)
    reset(stackfile);
    for j:=0 to 99 do
      begin;
        e:=(i*400)+(j*4);
        seek(stackfile,e);
        read(stackfile,h0);
        read(stackfile,h1);
        read(stackfile,h2);
        read(stackfile,h3);
        ch0[j]:=h0;
        ch1[j]:=h1;
        ch2[j]:=h2;
        ch3[j]:=h3;
      end;
    close(stackfile);
    if upcase(see)='Y' then
      begin;
        for k:=0 to 99 do               (*PRINT TO SCREEN*)
          begin;
            writeln(ch0[k]:5,'      ',ch1[k]:5,'      ',ch2[k]:5,
              '       ',ch3[k]:5);
          end;
      end;
```

```
(*  write ch0 & ch2 in two diff. files*)
  Assign(outfile,'power.dat');
  reset(outfile);
  e:=i*100;
  seek(outfile,e);
  for j:=0 to 99 do
    begin;
      r0:=ch0[j];
      write(outfile,r0);
    end;
  close(outfile);

  Assign(outfile,'posit.dat');
  reset(outfile);
  e:=i*100;
  seek(outfile,e);
  for j:=0 to 99 do
    begin;
      r2:=ch2[j];
      write(outfile,r2);
    end;
  close(outfile);

end;
Writeln('DONE--Press any key');
ch :=readkey;
goto outtahere;

(*SECTION 2  the INTERPOLATER*)

interpolate:
  begin;
  setgrowdelay(60);
  setshadowoff;
  settitlecenter;
  cyanblack;

  Growwindow(5,3,77,15,'Interpolater');
  Writeln('');
  writeln('I will assume that a high abs voltage=> RIO in positive position');
  writeln('I further assume that angular velocity is approx. constant when in
motion.');
  writeln('And I will assume that data was sampled at a steady rate of 1000hz/ chnl');
  Writeln(' ');
```

```
gotoxy(1,9);

Write('Where should I look for the data <CR> = "posit.dat" ');
readln(filebuff);
if filebuff='' then filebuff:='posit.dat';
Write('And where shall I put the output <CR> = "inputrx.dat" ');
readln(myfile);
if myfile='' then myfile:='inputrx.dat';

assign(outfile,myfile);
rewrite(outfile);
close(outfile);

ch:=readkey;
Writeln('Processing');

assign(interfile,filebuff);
reset(interfile);
fs:=filesize(interfile);
fsold:=fs;
close(interfile);

transit:=false; high:=false; low:=false; done:=false; killit:=false;
i:=0;   h0:=0;   h1:=0;   h2:=0; e:=0; c:=0; check:=0; where:=0;

(*is 1st point is in transit?*)
reset(interfile);
geth0;
checktrans;
if transit=false then h1:=0 else firsttrans;

while i < (fs-100) do            (*LOOP IS HERE*)
  begin;
    reset(interfile);
    seek(interfile,i);

  for e:=0 to 99 do
   begin;
    geth0;
    if h0<100 then
     begin;
      low:=true;
      high:=false;
      transit:=false;
```

```
            h1:=0;
          end;
      if h0>20000 then
       begin;
        h1:=1000;
        low:=false;
        high:=true;
        transit:=false;
       end;

      if (h0>100) and (h0<15000) then checktrans else transit:=false;
      if transit=true then lowinterpolate;
      if transit=false then ch0[e]:=h1;  (*This is h1 from the last point
                                    we hold h1 the same for glitches*)
     end;
    (*we get here if a) we made it thru above loop without a transit
                   b) if we had a transit*)
    if transit= false then
       begin;
        close(interfile);
        reset(outfile);
        seek(outfile,i);
        for l:=0 to 99 do
          begin;
            h2:=ch0[l];
            write(outfile,h2)
          end;
        i:=i+100;
       end;
     if killit=true then i:=fs;
    end;
  end;
(*PADDER*)     (*pads end with any necessary zeros to even file lengths*)
reset(outfile);
fs:=filesize(outfile);
if fs<fsold then
  begin;
   reset(outfile);
   seek(outfile,fs);
   for c:= fs to (fsold-1) do
     begin;
       h2:=0;
       write(outfile,h2);
     end;
```

```
    end;
  (*END PADDER*)

  goto outtahere;

outtahere:
  begin;
  clrscr;
  setgrowdelay(60);
  setshadowoff;
  settitlecenter;
  cyanblack;

  Growwindow(5,3,77,15,'We Are Done');
  Writeln('');
  write('Would you like another function? ');
  readln(ch);
  if upcase(ch) < > 'N' then goto mainloop;

  end;
end.
```

```
Program SneekPeek;
{by MIDN J C Ventura}

uses crt,dos, windows;

VAR
  myfile          :        string[12];
  filebuff        :        string[12];
  stackfile       :        file of integer;
  realfile        :        file of integer;
  h0,h1,h2,h3     :          integer;
  r0,r1,r2        :          integer;
  start,stop,step :        longint;
  add,add1,add2   :          longint;
  time            :        real;
  avg             :        longint;
  e,i,j,k         :        longint;
  FS,fnum         :          longint;
  see,ch          :        char;
  option          :        integer;

(* THE DATA COMPUTER ONLY GETS 1000 sample EACH channel per sec*)

label mainloop;
label punchout;
label showset;

Procedure Blackcyan;
  begin;    textcolor(black);    textbackground(cyan);    end;

Procedure redblack;
  begin;    textcolor(red);      textbackground(black);   end;

Procedure Cyanblack;
  begin;    textcolor(cyan);     textbackground(black);   end;



  begin;
  MAINLOOP:
   clrscr;
   setgrowdelay(60);
```

```
setshadowoff;
settitlecenter;
blackcyan;

(*SHOWSET ROUTINE!*)

showset:
begin;
 setgrowdelay(60);
 setshadowoff;
 settitlecenter;
 blackcyan;

Growwindow(5,3,77,15,'Sneekpeek on single channel data');
Writeln('');

writeln('What file would you like to read? ');
writeln('                              1. power.dat');
writeln('                              2. posit.dat');
writeln('                              3. pos1.dat');
writeln('                              4. inputrx.dat');
writeln('                              5. other file ');
write('                           6. END');
write('                           YOUR CHOICE:  ');
readln(option);
if option=1 then  myfile := 'power.dat';
if option=2 then  myfile := 'posit.dat';
if option=3 then  myfile := 'pos1.dat';
if option=4 then  myfile := 'inputrx.dat';

if option=6 then goto punchout;
if (option =5) then
  begin;
    write('What is the name of this  "other" file? ');
    readln(myfile);
    if myfile= ' ' then myfile:='power.dat';
  end;
if (option<1) or (option>6) then goto showset;
assign(realfile,myfile);
reset(realfile);
fs:=filesize(realfile);

 begin;
    writeln('This file has: ',fs,' data points.');
```

```
        writeln(' ');
        writeln('Start Range at? ');
        read(start);
        writeln('End Range at? ');
        read(stop);
        writeln('How many points should I average each time? ');
        read(avg);
        if  (avg=0) then avg:=1;

        if (stop=0) or (stop < start) then stop:= fs;
        Writeln('Step Size? ');
        read(step);

        if step=0 then step:=100;  (*100 points = .1 sec*)
        if stop>fs then stop:=fs;

        writeln('We will read from ',start,' to ',stop,' looking at every ',step,'
points.');
        writeln('Any key to continue');
        ch :=readkey;

    i:=start;
    while i < stop do
      begin;
       add:=0;
       for j:=0 to (avg -1) do
        begin;
         e:=i+j;
         seek(realfile,e);
         read(realfile,r0);
         add:=add+r0;
        end;
       add:=add DIV avg;
       time:=i / 1000;
       writeln(time:9:3,'      ',add:6);
       delay(200);
       i:=i + step;
      end;
    close(realfile);
   end;
  ch:=readkey;
end;
  goto mainloop;
punchout:
```

```
begin;
 clrscr;
 setgrowdelay(60);
 setshadowoff;
 settitlecenter;
 cyanblack;

 Growwindow(5,3,77,15,'We Are Done');
 Writeln('');

 write('Would you like another function? ');
 readln(ch);
 if upcase(ch) < > 'N' then goto mainloop;
 end;


end.
```